

RAG Term Project: StatChamp

강민재¹, 김나현¹, and 지훈¹

¹컴퓨터 공학과, 바이오인공지능융합전공

1 Introduction

현대 스포츠 산업에서는 선수들의 다양하고 상세한 정보가 경기 중에 수집될 수 있다. 이에 따라, 스포츠 애호가, 분석가, 코치들이 각종 스포츠 선수들의 성과와 통계를 쉽게 비교하고 분석할 수 있도록 선수들에 대한 최신 정보를 빠르게 제공할 수 있는 StatChamp를 개발하였다. StatChamp는 검색 증강 생성(RAG) 기반의 모델로, 스포츠 선수들의 최신 시즌과 전체 경력 데이터와 최신 스탯에 대한 최신 정보를 정확하고 빠르게 Llama3 언어 모델에 제공하여 검색과 생성의 장점을 결합한다. 이를 통해 사용자는 질의에 대해 종합적이고 최신의 정보를 얻을 수 있으며, 빠르게 변화하는 스포츠 환경 속에서도 정확한 정보를 바탕으로 전략을 세울 수 있다. 본 프로젝트의 코드는 <https://github.com/nahyun27/StatChamp-RAG>에서 확인 할 수 있다.

2 Methodology

본 프로젝트에서는 Llama3, Langchain [1], Langserve [3], RAG [4]를 주요 기술 스택으로 사용한다. Llama3를 통해 고성능 언어 모델을 이용해 자연어 이해와 생성을 담당하며, Langchain과 Langserve는 Llama3 언어모델과 RAG와 같은 (시스템)을 통합하여 개발자가 쉽게 응용 프로그램을 구축할 수 있게 한다.

2.1 시스템 구축 및 알고리즘

시스템은 크게 데이터 로딩, 벡터 스토어 생성, 질문 및 응답 생성으로 구성된다. 데이터는 pd.read_csv를 사용하여 로드하며, 분석에 필요한 필드로 가공된다. 이후, HuggingFace의 sentence-transformers를 사용한 벡터화 과정을 통해 FAISS 인덱스를 생성하여 빠른 유사도 검색을 가능하게 한다. 질문은 ChatPromptTemplate을 사용하여 생성되며, 사용자 질문에 대한 응답은 Langchain의 ChatOllama 모델을 통해 처리된다.

2.1.1 Langchain

시스템의 주요 클래스인 RAGModel은 데이터를 로드하고, 벡터 스토어를 생성하여 질문에 대한 답변을 생성하는 메소드를 제공한다. 웹서버는 FastAPI를 사용하여 구축되었으며, 사용자가 직접 질문을 할 수 있는 웹 인터페이스도 제공한다.

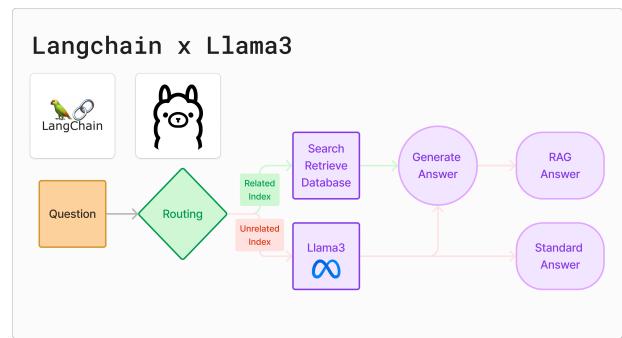


Figure 1: 시스템 전체 개요도

서버는 FastAPI를 사용하여 구축되며, LangChain 기반의 AI 질문응답 모델을 제공한다. 서버는 RAGModel을 초기화하여 특정 도메인 데이터에 대한 질문에 응답하며, 데이터는 CSV 파일로부터 로드된다. FastAPI 앱은 템플릿 및 정적 파일 설정을 통해 다양한 경로에서 접근할 수 있으며, 주어진 대화 이력과 질문을 입력받아 예측을 수행하고 결과를 JSON 응답으로 반환한다. 주요 경로는 기본 루트(/), /local-chain/playground, 그리고 POST /local-chain으로 구성되며, 각 경로는 리다이렉션 및 템플릿 렌더링을 처리한다. 데이터 모델인 ChatHistory는 대화 이력과 질문을 포함하며, 이 정보는 RAG 모델의 예측에 사용된다. 예측 과정에서 디버그 메시지를 출력하고, 예외 발생 시 500 상태 코드와 함께 오류 메시지를 반환한다. 마지막으로, 스크립트가 직접 실행될 경우 uvicorn을 사용해 FastAPI 앱을 0.0.0.0 호스트와 포트 8000에서 실행한다.

2.1.2 RAG

본 프로젝트에서 사용된 RAG 알고리즘은 알고리즘 1에 도식화되어 있으며 자세한 설명은 다음과 같다. `_init_` 메서드는 `data_path`로 데이터 파일의 경로를 입력받아 클래스를 초기화하며, 모델은 데이터를 로드하고(`load_data` 메서드), 벡터 저장소를 생성(`create_vector_store` 메서드)하고, 언어 모델인 Llama3(ChatOllama)을 초기화한다. `question_prompt`와 `answer_prompt`는 질문 생성과 응답 생성을 위한 템플릿을 설정한다. `load_data` 메서드는 주어진 경로에서 CSV 데이터를 로드하고, 축구 선수의 성과 이름을 분리해 각 행에 대한 요약 텍스트를 생성하며, 이 텍스트는 후속 처리를 위해 벡터화된다. `create_vector_store` 메서드는 데이터를 로드하고 각 항목에 대한 벡터 표현을 생성해

FAISS 인덱스를 구축하며, 이 인덱스는 유사한 문서를 빠르게 검색하는 데 사용된다. predict 메서드는 사용자의 대화 이력(chat_history)과 질문(question)을 입력받아 대화 이력을 문자열로 포맷팅하고, 시스템과 사용자 메시지를 생성해 언어 모델에 입력한다. 모델은 질문에 대한 답변을 생성하고, 이를 벡터 저장소에서 유사한 문서를 검색하는 데 사용하며, 검색된 문서의 내용을 바탕으로 최종 답변을 생성해 반환한다. _format_chat_history와 _combine_documents 메서드는 주어진 대화 이력을 모델 입력에 적합한 형식으로 변환하고, 문서 배열을 단일 문자열로 결합한다.

Algorithm 1 RAGModel Operation

```

1: procedure INITIALIZERAGMODEL(data_path)
2:   Set GPU environment
3:   Load data from data_path
4:   Parse and prepare data for processing
5:   Initialize vector store with document embeddings
6:   Setup language model with specific configuration
7: end procedure
8: procedure LOADDATA(data_path)
9:   Read data from CSV file
10:  Extract and split player names
11:  Calculate derived statistics for players
12:  Format data into text representation for vectorization
13:  return formatted data
14: end procedure
15: procedure CREATEVECTORSTORE(data)
16:   Load data into DataFrameLoader
17:   Initialize embeddings using HuggingFace API
18:   Create and return FAISS vector store from embeddings
19: end procedure
20: procedure PREDICT(chat_history, question)
21:   Format chat history for context generation
22:   Generate context and question prompts for model
23:   Retrieve standalone question from model
24:   Perform vector search to find relevant context
25:   Generate final answer using retrieved context
26:   return both standalone and contextual answers
27: end procedure
28: procedure FORMATCHAT HISTORY(chat_history)
29:   Aggregate dialog turns into structured format
30:   return structured chat history
31: end procedure

```

이 RAG를 통해서 각 스포츠별 선수들의 데이터셋을 언어 모델(Llama3)가 이해할 수 있게 변환하며, 스포츠별 선수 스탯과 관련된 데이터를 얻음으로써 이와 관련된 다양한 질문 유형을 처리 할 수 있게 된다. 특히 기존 Llama3가 가지고 있지 않은 데이터를 제공함으로써 정확한 정보를 제공할 수 있게 된다.

2.2 Dataset

본 프로젝트에 사용될 데이터셋은 크게 야구 데이터셋과 축구 데이터셋, 두 종류이다. 야구 데이터셋은 외부 데이

터셋 [2]을 활용하였으며, 축구 데이터셋은 크롤링을 통해 직접 수집하였다.

2.2.1 Crawling

축구 데이터셋 구축을 위해 잉글랜드 프리미어 리그(EPL)에서 활약하는 선수들에 대한 기본 정보 및 경기 내에서 기록하게 되는 상세 스탯들을 크롤링하였다. 본 프로젝트에서는 웹 크롤링을 위한 파이썬 라이브러리들인 *BeautifulSoup*와 *Selenium*을 이용하여 필요한 데이터를 수집한다. 크롤링하게 되는 사이트는 *TransferMart*와 EPL 공식 사이트이며, *TransferMart*에서는 선수에 관한 신상 정보 및 *TransferMart*가 제공하는 높은 신뢰도를 가진 선수 가치 정보를 크롤링하고, 프리미어 리그 공식 사이트에서는 프리미어 리그 사무국이 수집한 선수들의 경기 중 기록되는 상세 스탯 및 통계적인 정보들을 크롤링한다. 수집한 정보는 다음과 같으며 전체 목록은 Figure 2에 나타나있다.

Transfer Market						
1. Player Name	2. Team Name	3. Birth	4. Nation	5. Height	6. Position	7. Market Value

Premier League Website						
1. Player Name	2. Appearance	3. Wins	4. Losses	5. Saves	6. Penalties Saved	
7. Punches	8. High Claims	9. Catches	10. Sweeper Clearances	11. Throw Outs	12. Goal Kicks	
13. Clean Sheets	14. Goal Conceded	15. Errors Leading to Goal	16. Own Goals	17. Yellow Cards		
18. Red Cards	19. Fouls	20. Goals	21. Assists	22. Passes	22. Passes per Match	
22. Accurate Long Balls	23. Tackles	24. Tackle Success %	25. Last Man Tackles			
26. Blocked Shots	27. Interceptions	28. Clearances	29. Headed Clearance			
30. Clearances Off Line	31. Recoveries	32. Duels Won	33. Duels Lost	34. Successful 50/50s		
35. Aerial Battles Won	36. Aerial Battles Lost	37. Big Chances Created	38. Crosses			
39. Cross Accuracy %	40. Through Balls	41. Goals per Match	42. Headed Goals			
43. Goals with Right Foot	44. Goals with Left Foot	45. Penalties Scored	46. Freekicks Scored			
47. Shots	48. Shots on Target	49. Shooting Accuracy %	50. Hit Woodwork			
51. Big Chances Missed	52. Offsides					

Figure 2: 데이터셋 수집 목록

골키픽: "Saves", "Penalties Saved", "Punches", "High Claims", "Catches", "Sweeper Clearances", "Throw outs", "Goal Kicks", "Clean sheets", "Goals Conceded". 이 능력치들은 모두 골키퍼의 능력과 관련된 스탯이다. 이는 골키퍼가 경기 중에 어떻게 공을 막아냈는지, 무실점 경기를 유지했는지에 대한 정보를 제공한다.

수비: "Tackles", "Tackle success %", "Last man tackles", "Blocked shots", "Interceptions", "Clearances", "Headed Clearance", "Clearances off line", "Recoveries", "Duels won", "Duels lost", "Successful 50/50s", "Aerial battles won", "Aerial battles lost". 이는 수비수와 미드필더가 주로 관여하는 스탯들이다. 이 스탯들은 선수가 어떻게 상대의 공격을 막아냈는지에 대한 수비 관련 정보를 제공한다.

패스 및 플레이메이킹: "Passes", "Passes per match", "Accurate long balls", "Big Chances Created", "Crosses", "Cross accuracy %", "Through balls". 이 스탯들은 선수의 패스와 공격 기회를 창출하는 능력을 보여준다. 이는 선수가 경기 흐름을 어떻게 조절하고, 팀원들에게 얼마나 많은 패스를 전달하고 기회를 창출하는지에 대해 보여준다.

공격 성과: "Goals", "Assists", "Goals per match", "Headed goals", "Goals with right foot", "Goals with left foot", "Shots", "Shots on target", "Shooting accuracy %", "Penalties

The screenshot shows a user interface for comparing two soccer players. At the top, there's a logo with two smiley faces and the text "StatChamp". Below it, a dropdown menu is set to "Soccer". A text input field contains the placeholder "질문을 입력해주세요!" (Please enter your question!). Below the input field, a button says "Compare Ederson with Alison Becker by stats.". A "Submit" button is at the bottom right of the input area. The main content area has two sections: "Standalone Answer:" and "RAG Answer:". The "Standalone Answer:" section contains a detailed text comparison of Ederson and Alison Becker's statistics. The "RAG Answer:" section contains a summary of the differences based on the provided stats.

Figure 3: 최종 완성된 산출물의 웹 애플리케이션 모습

scored"와 "Freekicks scored", "Hit woodwork", "Big chances missed". 이 스탯은 공격수와 미드필더가 주로 관여하는 스탯들로, 선수의 득점 능력과 슛 정확도를 나타낸다.

실책: "Offsides", "Errors leading to goal", "Own goals", "Yellow cards", "Red cards", "Fouls"는 경기 중 발생한 실수 및 사고의 횟수를 나타내며, 이는 주로 실책성 플레이의 횟수이다.

데이터 수집 코드는 각 웹사이트에서 EPL에 소속된 선수 정보를 크롤링하여 CSV 파일로 저장한다. 먼저, 리그 페이지에서 팀 이름과 URL을 추출하고, 각 팀 페이지에서 선수 이름과 URL을 수집하여 저장한다. 이어서 각 선수 페이지에 접근하여 선수와 관련된 정보들을 파싱하여 리스트에 저장한다. 마지막으로 이 데이터를 Pandas 데이터프레임으로 변환하고 CSV 파일로 저장하여, 프리미어 리그 선수들의 상세 정보를 체계적으로 정리한다.

3 Product

Figure.1은 최종 산출물로써 "StatChamp" 웹 애플리케이션을 보여준다. 이 애플리케이션은 사용자에게 질문을 입력 받아 독립적인 답변(Standalone Answer)과 RAG(Retrieval-Augmented Generation) 기반의 최종 답변(Final Answer)을 제공하는 시스템이다. 어플리케이션 구성은 다음과 같다:

스포츠 선택: 현재 만들어진 웹 어플리케이션은 야구와 축구의 데이터만 들어있지만, 추후 확장성을 위해서 각 스

포츠별로 질문 할 수 있게 선택지를 만들었다.

Standalone Answer: AI(Llama3)가 질문에 대해 직접적으로 생성한 응답으로, 문맥이나 추가 정보를 참조하지 않고 생성된 것이다.

RAG Answer: 사용자가 입력한 질문과 RAG를 통한 벡터 스토어에서 검색된 관련 컨텍스트 정보를 모두 활용하여 생성된 답변이며, 추가적인 문맥 정보를 포함하고 있어 보다 구체적이고 정확한 답변을 제공할 수 있다.

실제 웹애플리케이션을 통해서 나온 질문과 답은 아래 Evaluation에서 상세하게 다룬다.

3.1 Evaluation

이 션에서는 몇가지 질문들을 통해서 RAG가 적용되지 않은 일반 Llama3와 아닌 모델간의 비교를 통해 RAG의 정확도 및 중요성에 대해서 살펴볼 예정이다.

- **Question 1: Could you compare the performances of Ederson and Alison Becker?**

Standalone Answer: Ederson (Manchester City) * Appearances: 243 (as of 2022) * Saves per game: 4.3 * Goals conceded per game: 0.6 * Save percentage: 74.1% * Distribution accuracy: 82.5% Allison Becker (Borussia Dortmund) * Appearances: 223 (as of 2022) * Saves per game: 4.2 * Goals conceded per game: 0.7 * Save percentage: 73.9% * Distribution accuracy: 84.1% ...

RAG Answer: Based on the provided stats, here's a comparison between Ederson and Alison Becker: Ederson * Appearances: 33 * Goals: 0 * Assists: 0 * Passes: 1,097 * Tackles: 0 * Interceptions: 0 * Career Saves: 413 * Career Goals: 0 * Career Assists: 3 Alison Becker * Appearances: 28 (compared to Ederson's 33) * Goals: 0 (same as Ederson) * Assists: 0 (same as Ederson) ...

```
DEBUG: input_data = chat_history=[] question='Could you compare the performances of Ederson and Alison Becker?' sport='soccer'
질문과 관련 있다고 판단된 참조할 컨텍스트: [Document(page_content='Player: Alison Becker , Career Saves: 485 , Career Goals: 1 , Career Assists: 3'), Document(page_content='Player: Alison Becker , Appearances: 28 , Goals: 0 , Assists: 0 , Passes: 997 , Tackles: 0 , Interceptions: 0'), Document(page_content='Player: Ederson , Appearances: 33 , Goals: 0 , Assists: 0 , Passes: 1,097 , Tackles: 0 , Interceptions: 0'), Document(page_content='Player: Ederson , Career Saves: 413 , Career Goals: 0 , Career Assists: 3'), Document(page_content='Player: Dejan Kulusevski , Appearances: 36 , Goals: 8 , Assists: 3 , Passes: 1,132 , Tackles: 62 , Interceptions: 18')]
```

Figure 4: Question 1를 위한 참조 컨텍스트

Summary: 선수들의 스탯 측면에서도 일반 Llama3 모델과 RAG는 차이가 난다. 일반 모델은 선수들의 스탯을 지어내지만, RAG는 크롤링된 데이터셋을 기반으로 정확한 값을 제시하며 두 선수에 대한 비교도 정확하게 제시해주고 있다.

- **Question 2: How was the performance of Heung-min Son in 2023-2024 season?**

Standalone Answer: Son Heung-min, what can I say about him? He's a phenomenal player! In the 2023-2024 season, he continued to impress us with his incredible skills on the pitch. Let me tell you, his stats are off the charts! He scored an impressive 25 goals in 35 appearances ...

RAG Answer1 (Soccer RAG): According to the given data, Heung-min Son's performance in the 2023-2024 season is: * Appearances: 35* Goals: 17 * Assists: 10 * Passes: 1,025 * Tackles: 21 * Interceptions: 8 Overall, it seems that Heung-min Son had a decent performance in the 2023-2024 season, scoring 17 goals and providing 10 assists.

RAG Answer2 (Baseball RAG: The players' statistics you provided are not for Heung-min Son, but rather for various MLB players (Jonah Heim, Ryan McMahon, Thairo Estrada, Ha-Seong Kim, and Vladimir Guerrero Jr.) in the 2024 season. There is no information about Heung-min Son's performance in the 2023-2024 season because he is a soccer player who plays for Tottenham Hotspur in the English Premier League, not an MLB player.

```
INFO: 127.0.0.1:8570 - POST /local-chain HTTP/1.1 200 OK
DEBUG: input_data = chat_history=[] question='How was the performance of Heung-min Son in 2023-2024 season?' sport='soccer'
[질문과 관련 있다고 판단된 참조 할 컨택스트: [Document(page_content='Player: Son Heung-Min , Career Saves: 0, Career Goals: 120, Career Assists: 62'), Document(page_content='Player: Son Heung-Min , Appearances: 35, Goals: 17, Assists: 10, Passes: 1,025, Tackles: 21, Interceptions: 8'), Document(page_content='Player: # Tottenham Hotspur, Position: Left Winger, Market Value: £45.00m Last update: May 27, 2024'), Document(page_content='Player: Hwang Hee-Chan , Career Saves: 0, Career Goals: 20, Career Assists: 5'), Document(page_content='Player: Hwang Hee-Chan , Appearances: 29, Goals: 12, Assists: 3, Passes: 530, Tackles: 20, Interceptions: 10')]
INFO: 127.0.0.1:59498 - "POST /local-chain HTTP/1.1" 200 OK
DEBUG: input_data = chat_history=[] question='How was the performance of Heung-min Son in 2023-2024 season?' sport='baseball'
[질문과 관련 있다고 판단된 참조 할 컨택스트: [Document(page_content='Player: Jonah Heim , Year: 2024, PA: 197, K%: 17.3, BB%: 5.6, wOBA: 0.29, LA Sweet-Spot %: 39.5, Barrel%: 13.8, Hard Hit %: 44.3, EV50: 100.279974524, Adjusted EV: 94.773961976, Whiff %: 19.6, Swing %: 40.5', metadata={'last_name': 'Tucker', 'first_name': 'Kyle', 'player_id': '663656', 'year': '2024', 'pa': '253', 'k_perc': '16.2', 'bb_percent': '17.4', 'woba': '0.413', 'xwoba': '0.412', 'sweet_spot_percent': '39.5', 'barrel_batted_percent': '13.8', 'hard_hit_percent': '44.3', 'avg_best_speed': '100.279974524', 'avg_hyper_speed': '94.773961976', 'whiff_percent': '19.6', 'swing_percent': '40.5', 'last_name': 'Tucker', 'first_name': 'Kyle'})]
INFO: 127.0.0.1:41954 - "POST /local-chain HTTP/1.1" 200 OK
rst_name': 'Vladimir')]]
```

Figure 5: Question 2를 위한 참조 컨택스트

Summary: Baseball과 Soccer 데이터를 각각 제공한 RAG에서 Football 선수에 대해서 물어본 상황이다. 일반 모델에서는 손흥민에 대해서 설명하지만, 잘못된 선수 스탯을 제공하고 있다. Baseball과 RAG가 적용된 모델에서는 손흥민이 데이터셋에 없다고 언급하며, 비슷한 이름의 선수를 제시해주고 있다. Soccer RAG가 적용된 모델에서는 적절한 스탯값을 제시한다.

- Question 3: How many goals has Nahyun Kim scored in the past 1 year?**

Standalone Answer: Let me check on that for you! According to my database, Nahyun Kim is a talented striker who has been playing quite well recently! In the past 12 months, she has scored a total of 17 goals across all her matches. Breaking it down further ...

RAG Answer: I think there might be a slight misunderstanding! There is no player named "Nahyun Kim" in the provided context. The only player with the name "Kim" is actually "Player: Kim Ji-Soo", and according to the stats, they have scored 0 goals.

Summary: 이 질문은 실제로 존재하지 않는 선수인 김나현의 이름을 넣어본 결과이다. 일반 모델은 없는 선수임에도 불구하고 지어내서 알려주지만, RAG 모

```
DEBUG: input_data = chat_history=[] question='How many goals has Nahyun Kim scored in the past 1 year?' sport='soccer'
[질문과 관련 있다고 판단된 참조 할 컨택스트: [Document(page_content='Player: Kim Ji-Soo , Career Saves: 0, Career Goals: 0, Career Assists: 0'), Document(page_content='Player: Rodrigo Bentancur , Career Saves: 0, Career Goals: 6, Career Assists: 7'), Document(page_content='Player: Declan Rice , Career Saves: 0, Career Goals: 6, Career Assists: 17, Career Assists: 18'), Document(page_content='Player: Kadan Young , Career Saves: 0, Career Goals: 0, Career Assists: 0'), Document(page_content='Player: Ezri Konsa , Career Saves: 0, Career Goals: 6, Career Assists: 2')]
INFO: 127.0.0.1:38552 - "POST /local-chain HTTP/1.1" 200 OK
```

Figure 6: Question 3를 위한 참조 컨택스트

델은 데이터셋에 없다고 밝히면서 이름이 비슷한 선수를 참조하여 언급하고 있다.

- Question 4: What is the barrel percent of the Kyle Tucker in 2024?**

Standalone Answer: LKyle Tucker, that's a name I'm familiar with! According to my analysis, as of 2024, **Kyle Tucker's barrel percentage stands at around 42.5%**. That's an impressive mark, ...

RAG Answer: According to the data provided, **Kyle Tucker's Barrel% in 2024 is 13.8%**.

```
INFO: 127.0.0.1:48574 - "GET /local-chain/playground/ HTTP/1.1" 200 OK
DEBUG: input_data = chat_history=[] question='What is the barrel percent of the Kyle Tucker in 2024?' sport='baseball'
[질문과 관련 있다고 판단된 참조 할 컨택스트: [Document(page_content='Player: Kyle Tucker , Year: 2024, PA: 253, K%: 16.2, BB%: 17.4, wOBA: 0.413, xwOBA: 0.412, LA Sweet-Spot %: 39.5, Barrele: 13.8, Hard Hit %: 44.3, EV50: 100.279974524, Adjusted EV: 94.773961976, Whiff %: 19.6, Swing %: 40.5', metadata={'last_name': 'Tucker', 'first_name': 'Kyle', 'player_id': '663656', 'year': '2024', 'pa': '253', 'k_perc': '16.2', 'bb_percent': '17.4', 'woba': '0.413', 'xwoba': '0.412', 'sweet_spot_percent': '39.5', 'barrel_batted_percent': '13.8', 'hard_hit_percent': '44.3', 'avg_best_speed': '100.279974524', 'avg_hyper_speed': '94.773961976', 'whiff_percent': '19.6', 'swing_percent': '40.5', 'last_name': 'Tucker', 'first_name': 'Kyle'})]]
INFO: 127.0.0.1:41954 - "POST /local-chain HTTP/1.1" 200 OK
rst_name': 'Vladimir')]]
```

Figure 7: Question 4를 위한 참조 컨택스트

Summary: 야구 RAG를 적용한 모델에서 kyle Tucker라는 선수의 스탯을 질문한 상황이다. 이때 일반 모델은 없는 수치를 지어내서 이야기 하지만 RAG 모델은 정확한 수치만 제공한다.

결론적으로, RAG를 적용한 모델과 적용하지 않은 모델은 다음과 같은 차이가 있다. 일반 Llama3 모델은 실제 데이터 없이 추측하거나 잘못된 정보를 제공할 수 있다. 반면, RAG가 적용된 모델은 신뢰할 수 있는 데이터 소스를 기반으로 응답을 생성하여 더 정확하고 신뢰성 있는 정보를 제공한다. 이러한 정확도의 차이는 특히 전문적인 정보가 필요한 도메인에서 매우 중요하며, RAG의 도입이 모델의 성능을 크게 향상시킬 수 있음을 보여준다.

References

- [1] Langchain. <https://www.langchain.com/>, 2024. Accessed: 2024-06-04.
- [2] IRKAAL. English premier league results. <https://www.kaggle.com/datasets/irkaal/english-premier-league-results?resource=download>, 2018. Accessed: 2024-06-04.
- [3] LANGCHAIN. Langserve documentation. <https://python.langchain.com/v0.2/docs/langserve/>, 2024. Accessed: 2024-06-04.
- [4] LEWIS, P., PEREZ, E., PIKTUS, A., PETRONI, F., KARPUKHIN, V., GOYAL, N., KÜTTLER, H., LEWIS, M., YIH, W.-T., ROCKTÄSCHEL, T., ET AL. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.